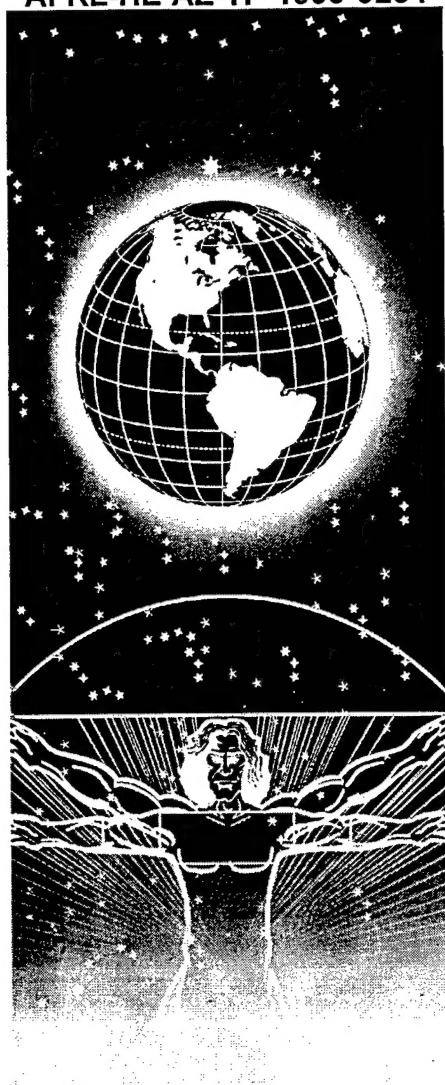


AFRL-HE-AZ-TP-1999-0254



UNITED STATES AIR FORCE RESEARCH LABORATORY

VISUAL INTERFACE UNIT: A MODERN APPROACH TO INTEGRATING IMAGE GENERATORS IN A DIS AND HLA ENVIRONMENT

Lance R. Call

L3 Communications
6030 S. Kent Street, Bldg 560
Mesa, AZ 85212-6061

20010703 067

July 2000

Approved for public release; distribution is unlimited.

AIR FORCE MATERIEL COMMAND
AIR FORCE RESEARCH LABORATORY
HUMAN EFFECTIVENESS DIRECTORATE
WARFIGHTER TRAINING RESEARCH DIVISION
MESA, AZ 85212-6061

NOTICE

Publication of this paper does not constitute approval or disapproval of the ideas or findings. It is published in the interest of scientific and technical information (STINFO) exchange.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Office of Public Affairs has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.

PHILIPP W. PEPPLER
Project Engineer

DEE ANDREWS
Technical Director

JERALD L. STRAW, Colonel, USAF
Chief, Warfighter Training Research Division

Please do not request copies of this report from the Air Force Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

Federal Government agencies and contractors registered with the Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
8725 John J. Kingman Road, Suite 0944
Ft. Belvoir, VA 22060-6218

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE July 2000	3. REPORT TYPE AND DATES COVERED Final - April 99 to Jan 00		
4. TITLE AND SUBTITLE Visual Interface Unit: A Modern Approach to Integrating Image Generators in a DIS and HLA Environment		5. FUNDING NUMBERS C - F4124-97-D-5000 PE - 62205F PR - 1123 TA - B1 WU - 04		
6. AUTHOR(S) Call, Lance R.		8. PERFORMING ORGANIZATION REPORT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) L3 Communications 6030 South Kent Street, Bldg 560 Mesa AZ 85212-6061		10. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-HE-AZ-TP-1999-0254		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Human Effectiveness Directorate Warfighter Training Research Division 6030 South Kent Street, Bldg 558 Mesa AZ 85212-6061		11. SUPPLEMENTARY NOTES Air Force Research Laboratory Technical Monitor: Mr Philipp W. Peppler, (480)988-6561 X-273, DSN 474-6273.		
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) This paper explores how simulators and image generators (IGs) may be integrated and presents a historical perspective of how hosts and IGs have traditionally been connected. How a Visual Interface Unit (VIU) may be used to integrate several different kinds of hosts (F-16, A-10, etc.) with multiple kinds of IGs while lowering the host/IG integration, maintenance, and upgraded costs is also discussed. An architecture using a Standard Host IG interface and a "network-enabled" IG capable of connecting to a "public" network to get most of its information is suggested. A network-enabled IG significantly reduces the amount of information that must be exchanged between a host and its IG. Using the Standard Host IG interface protocol would allow a "plug-and-plan" IG connection regardless of the vendor.				
14. SUBJECT TERMS Distributed Interactive Simulation; DIS; High level architecture; HLA; Host/IG integration; IGs; Image generators; Visual displays; Visual interface units; VIU			15. NUMBER OF PAGES 16	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION ABSTRACT Unlimited	

PREFACE

The work discussed in this paper was performed in support of the Air Force Research Laboratory, Human Effectiveness Directorate, Warfighter Training Research Division (AFRL/HEA) in Mesa AZ. It was conducted to find a way to efficiently integrate a variety of aircraft simulators (F-16, A-10, T-38, C-130, etc.) with many different image generators (IGs), including Lockheed Martin, Evans & Sutherland (E&S), Silicon Graphics Inc. (SGI), and PC-based.

The work was conducted under Work Unit 1123-B1-04, Visual Training Research; Contract Number F41624-97-D-5000 with L-3 Communications (formerly Raytheon Training Services). The laboratory contract monitor was Mr Philipp W. Pepler, AFRL/HEAE, and the laboratory work unit monitor was Dr Byron J. Pierce, AFRL/HEAE.

VISUAL INTERFACE UNIT: A MODER APPROACH TO INTEGRATING IMAGE GENERATORS IN A DIS AND HLA ENVIRONMENT

Introduction

The Air Force Research Laboratory, Human Effectiveness Directorate, Warfighter Training Research Division (AFRL/HEA) in Mesa AZ is involved in research, development, and demonstration of technology that can be used to train warfighters. One focus of the laboratory is to create affordable and accurate flight simulators that may be used in a Distributed Mission Training (DMT) environment to train pilots. The laboratory has built several different types of simulators including F-16, A-10, C-130, and T-38 flight simulators.

Another focus of the laboratory is to explore new and improved visual system technology. This includes displays, display layouts, display materials, and super high-resolution projectors. It also includes the computer technology used to create out-the-window image displays. This device is often referred to as an image generator (IG). The image generator may be a dedicated hardware/software combination box as has been the case with Lockheed-Martin and Evans and Sutherland products, or it may use a general purpose computer such as a Silicon Graphics or a PC running a software package such as Aechelon, Mak Technologies SimView, SDS International Inc., LiteFlite, Powerscene, or Top Scene.

For a simulator host to use an IG, they must be integrated in some way. AFRL/HEA provides an opportunity to explore the host/IG integration because it has many different kinds of cockpits and image generators from several major vendors. We often need to connect cockpits to different IGs and also the same IG to different cockpits. This provides the opportunity to explore how to deal with a variety of host/IG combinations. We also have a working relationship with the IG manufacturers that allows us to explore, implement, and test new concepts.

This paper explores how simulators and image generators may be integrated and presents a historical perspective of how hosts and IGs have traditionally been connected. I will discuss how a Visual Interface Unit (VIU) may be used to integrate several different kinds of hosts (F-16, A-10, etc.) with multiple kinds of IGs while lowering the host/IG integration, maintenance, and upgraded costs. Also I will suggest an architecture using a Standard Host IG interface and a "network-enabled IG" capable of connecting to a "public" network to get most of its information. A network-enabled IG significantly reduces the amount of information that must be exchanged between a host and its IG. Using the Standard Host IG interface protocol would allow a "plug-and-play" IG connection regardless of the vendor.

Traditional Host IG Connection

The traditional way for a host to connect to an IG is by using the Interface Control Document (ICD) each IG manufacturer provides. Each manufacturer uses its own ICD which typically is proprietary. Each connection between a host and an IG required a custom software integration that was usually done by the host to conform to the IG's ICD. If a new IG was to be used, then the old interface would be scrapped and new software would be generated to complete the interface. For example this would be the case if an F-16 simulator were switched from an Evans and Sutherland IG to a Lockheed Martin IG. If multiple types of hosts were connected to the same type of IG, there would also be unique software for that host. This would be the case when both an F-16 and an A-10 were connected to SGI machines using the same IG software. This puts the responsibility of host/IG integration on the host.

Typically, there was no external network to connect simulators and share information--the host created all simulation information (Figure 1). Hosts often developed a way to generate targets and other visual items and sent them to the image generator. Implementation was often very host specific. When enhancements, changes, or fixes were made to an interface, they would be hand integrated into the interfaces of other hosts. Changes in the A-10's visual interface must be hand coded in the F-16, T-38, and C-130 interface. Many times improvements made for one host were not integrated into another due to time/manpower

constraints; therefore, over time, the code becomes more and more host specific, making it more and more difficult to copy improvements from one visual interface to another. Every interface must be debugged individually. This makes integrating, improving, and maintaining the host/IG interfaces very tedious work.

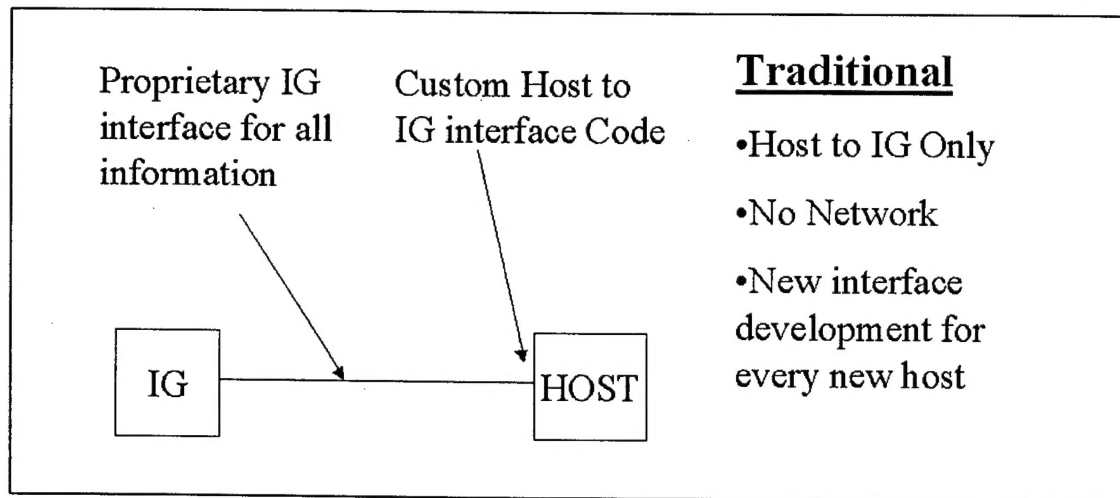


Figure 1. Typical Host-IG Connection

The Advent of Networked Simulators

A major occurrence was the use of common network standards such as SIMNET, Distributed Interactive Simulation (DIS), and High-Level Architecture (HLA), which allowed various simulators to be interconnected. Now information about other aircraft, tanks, etc., was likely to be coming from another simulator or computer-generated force (CGF) model on the network (Figure 2). The information was not being generated by the host as it previously had been. When this occurred, all of the AFRL/HEA simulators took the following approach to being integrated with the network.

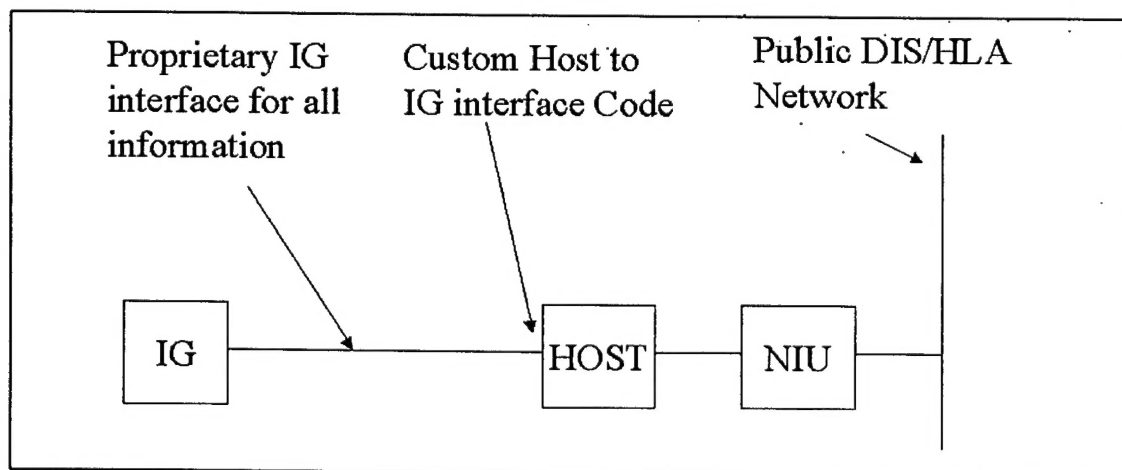


Figure 2. Common Networked Host-IG Interface

1. They continued to use the same IG interface that they had used when they were connected as standalone devices.
2. A Network Interface Unit (NIU) was used to translate network data into a common format to be used by all hosts. All hosts could write to a common ICD and then be able to send and receive data. All

types of hosts would use the same NIU code making it easier to maintain, update, test, and debug the NIU. The hosts took the information from the NIU and filled in their already existing variables for targets, and so forth, that they had used internally when they were standalone devices. This made integration with the aircraft systems and the IG (since the interface was already done) easier. This worked, but did not address any of the integration, maintenance, consistency of operations/effects between hosts, or improvement issues in the host/IG interface.

In addition, the individual limits of a simulation system were imposed on the visual. For example, the aircraft code in the F-16 was limited to 25 total targets. Because the network information flowed through the cockpit, the IG was also limited to 25 targets. Again the limitations of each host were different causing some differences in the visual interface to exist.

Host IG Connection Using the Visual Interface Unit

To address some of the problems with integration and maintenance, it would be desirable to have a single standard format interface that all hosts could use to communicate with all image generators. This obviously would not be the case with the IGs that AFRL/HEA already had. In looking at the architecture, we noticed that most of the data that the IG needed, such as host position and all target information, were coming from or being sent to the NIU in a standard format for all simulators. We came up with the concept of a Visual Interface Unit. The VIU would use the data being sent to and from the NIU and translate it into the various formats that the IG manufacturers expected (Figure 3). This could solve several problems:

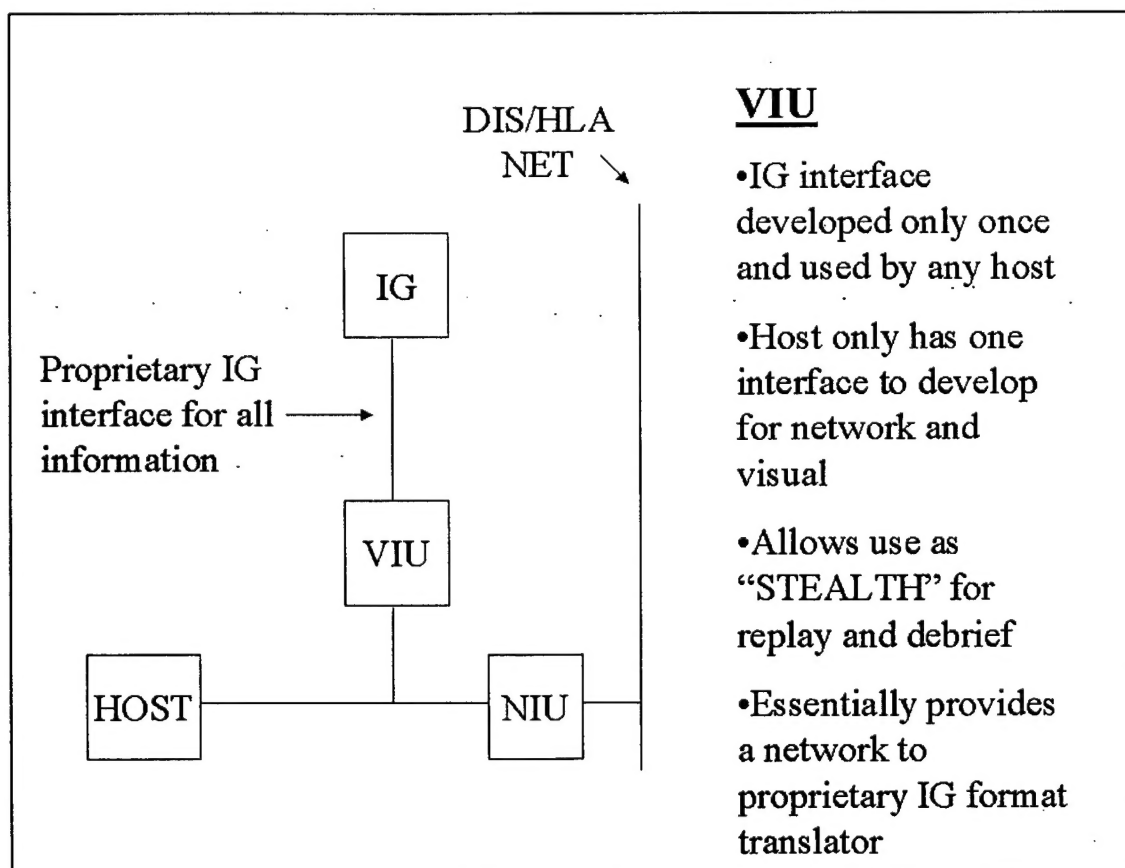


Figure 3. Interfacing the Host and IG with a VIU

1. From a host perspective the interface to the IG would always be the same regardless of what kind of IG was connected.
2. All different types of hosts could use the same interface.
3. The interface from a host to the network would also serve as the visual interface.
4. Only one version of the VIU need be implemented, maintained, and improved. Any improvements and fixes would apply to all hosts using the VIU. The visual interface would keep IGs of the same type consistent with each other regardless of which host they were connected.
5. The interface to a new IG would only have to be implemented once, and would then be available to all hosts.
6. New hosts would only have to implement the network interface and would then automatically have the visual interface done. The visual interface would be for all IGs that the VIU supported not just one IG.
7. Debugging and checkout of the interface could be done with any available cockpit and would apply to all hosts, rather than checking each one out individually for every change.

In addition to solving several problems, this architecture also allowed us several novel benefits. To understand some of these, it is necessary to understand a few of the implementation details of the VIU. The NIU buffers have entities that represent the host as well as the other aircraft, tanks, etc., that are on the network. The host and other entities are identical in format. The VIU is configured to "attach" to a particular entity in the buffers. This is just what a "Stealth" viewer does in a DIS exercise. This is normally the entity that the host is providing on the network. The VIU then sends this position along to the visual. This makes the visual interface essentially a "Stealth" that may be attached to any entity to see what that entity would be seeing. If the DIS data is logged during an exercise and then played back, the NIU will get the data and provide it to the VIU. The cockpit can be put into a "Stealth" mode, which will follow along with its data coming in from the network from the logger. This allows a pilot to fly a mission live and then sit in the cockpit and watch an instant replay of all the things he saw in his visual. This could be a valuable tool for debriefing a pilot. It has been a valuable tool to demonstrate an exercise months after the exercise was performed.

In addition to displaying host information, an IG often provides data to the host. This data may include terrain heights, ground-clamped positions, collision events, intervisibility, type of surface, etc. The VIU translates this data into a common format and appends or modifies the data in the NIU buffer before the host is allowed to read the buffer. This is the communications path from the IG to the host. The VIU makes all IGs appear the same to the host. The current version of the VIU supports terrain heights, ground-clamped positions, and ground-collision events. Intervisibility and type of surface are not used by the AFRL/HEA simulators and therefore were not implemented. They could however easily be added to the VIU scheme to provide these services.

The VIU is essentially a protocol translator. By having a common interface on the host side, the VIU is able to be a translator for any host. Although implemented, in this case, with the AFRL/HEA NIU, this concept may be used with other software. Many of the commercial NIU vendors take network data and convert it to a common backend Application Programming Interface (API). From this common backend, a VIU could be created to read and write to currently existing IG interface protocols. Using the backend of the NIU allows a host to operate in a "Stealth" mode for debrief, demonstration, and evaluation purposes. The main principles are that to the host the interface is always the same regardless of which IG is connected. The VIU reads and writes that common interface and converts it into the particular IGs interface protocol. The VIU is the IG device driver.

AFRL/HEA implemented the VIU and has been successfully using it at the laboratory for quite some time. It was integrated into the F-16, A-10, and T-38 simulators and used in the Roadrunner 98 training missions using both SGI and Lockheed Martin's CompuScene 2000+ image generators. It was used also at the 1998 Interservice/Industry Training, Simulation, and Education Conference (IITSEC 98) to demonstrate a high-

resolution satellite photography database of the Nellis range complex and a multispectral night vision database. Both databases were built by Aechelon and ran Aechelon software on an SGI. The VIU was also used to demonstrate two different databases running the MultiGen/Paradigm software on the SGI. We have realized all the benefits that we thought we would see by implementing the VIU concept.

VIU with a Network-Enabled IG

Conceptually, the IG should know about all the targets, effects, weather, time of day, etc., in the environment and everything in the virtual environment that may be visualized. These are independent of the host, meaning that an A-10 or an F-16 should see the same thing if they were in the same place. The view does not change because of the host. The host just controls the positioning of the viewpoint in the virtual world. If this information is independent of the host, why should the host be sending it to the IG? The host no longer creates most of the information that makes up the virtual world. The information is created by a network of simulations, and simulators that are sharing their information over some type of public network. Today this is likely to be a DIS or HLA network. The IG could get the information about targets, effects, weather, time of day, etc., directly from the network. In the previous example, the NIU/VIU pair was required to translate all network information into the ICD format used by the particular IG manufacturer. This essentially provided a network interface for the IG, but requires at least two layers of translation, which adds complexity, delay, and latency. If the IG had its own NIU, it could access the "public" DIS or HLA network directly. It could access all network data without translation (Figure 4). The IG manufacturer could use any network information to help in the visualization. They could implement the effects to highlight the particular capabilities of their IG rather than hoping that the integration contractors would implement them in the host-to-IG interface. It is likely that the IG manufacturer knows the IG best. Having a "network-enabled" IG allows the person who knows the IG best to implement the visualization of the world. The host would just need to tell the IG what view of the world it would like to see.

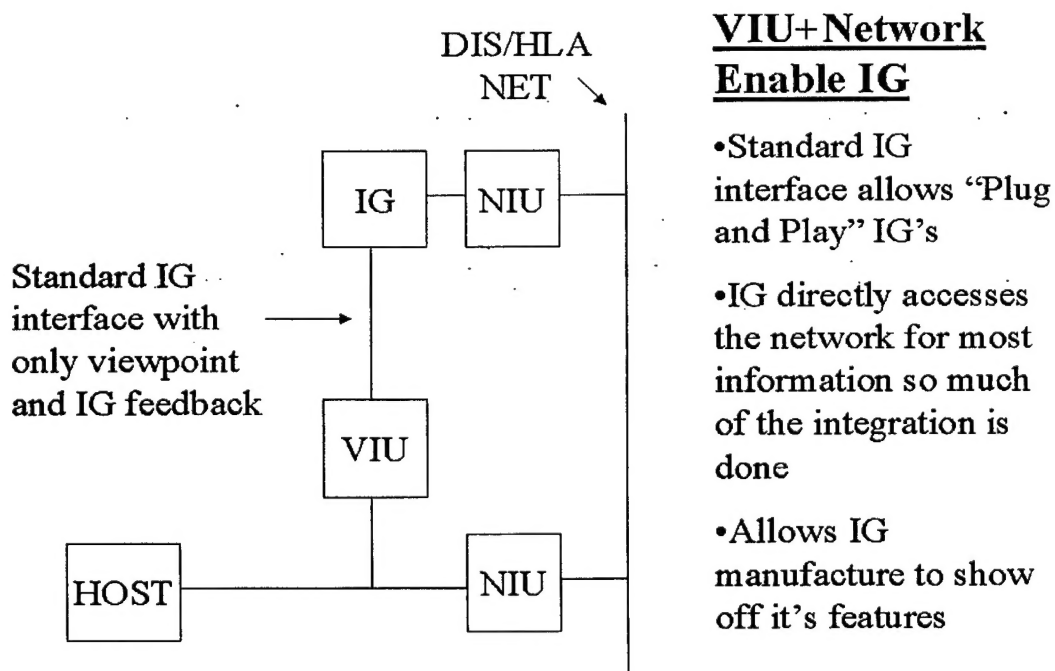


Figure 4. The Standard IG (STDIG) Interface

At first it might seem that all the information an IG would need is available from the public network making an IG simply a "stealth." There are a few concerns about this.

First, data on the public network is updated only when necessary to reduce network traffic. This leads to latency and some jumping in the rotations that are particularly noticeable in the visualization. It is much more disturbing when the whole horizon line moves due to a network update than when a model that takes up only a small portion of the screen jumps or rotates due to a network update. Smoothing applied to make network updates smooth result in additional latency. In informal tests we used DIS with various error thresholds, and with and without smoothing, to see if a "Stealth" connected only to the public network would be sufficient. The results are unacceptable. The latency observed is too disturbing to the pilots. Only when the data were sent out essentially every frame was the visual acceptable. This would quickly flood a public network with data and is therefore unacceptable.

Second, the IG often needs to provide data to the host that is specific to the host. This information does not apply to the public at large and therefore it would be better not to add it to the public network load. For these reasons there should be two networks. One "PUBLIC" network from which the IG gets most of its data, and one "PRIVATE" network over which the host and IG communicate every frame in order to reduce latency.

Since a network-enabled IG gets most of its information from the public DIS/HLA network, the information required from the host is greatly simplified. In its simplest form it is just the position of the host that needs to be provided to the IG. A VIU can still be used to translate host information into the format required by the IG, so a VIU is still desirable. It is desirable to have only one format of data for all IGs. With a network-enabled IG, this is quite possible. AFRL/HEA is exploring a Standard IG protocol interface that defines the formats of the data that may be sent or received from a network-enabled IG. A draft standard has been proposed. An initial test has been done with Multi-Gen/Paradigm on our SGI platform. Aechelon and Mäk Technologies are also planing to implement this Standard IG interface protocol. The protocol is currently open for comments and suggestions. By only supporting network-enabled IGs, the protocol is kept simple. Model-type translations, animations versus states, chaining of models, and other issues associated with visualizations of models that are different for different IGs are not an issue. This is because the IG gets model information from the public network, not through the Standard IG interface. The Standard IG interface uses the following messages:

1. BEGIN_FRAME (REQUIRED) – indicates the beginning of a frame.
2. END_FRAME (REQUIRED) – indicates the end of a frame.
3. VIEWPOINT_POSITION (REQUIRED) – gives the position and direction of the viewpoint.
4. VIEWPOINT_DYNAMICS – if necessary provides velocities to allow extrapolation.
5. ENUMERATED_DATA – allows hosts to send data by type to the IG, flexible way to add future capabilities. This is not needed at this point.
6. FIELD_OF_VIEW – defines the viewing frustums for each window.
7. WINDOW_CONTROL – allows controls required by sensor channels, such as WHITE-HOT etc.
8. FILTER_OBJECT – allows filtering of specific objects. Used to filter the host from the public data.
9. EVENT_NOTIFICATION – used by the IG to send information to the host about terrain height, collisions, ranges, etc.

This protocol would allow IGs to be "plug-and-play" for hosts that have a Standard IG interface capability. Since most of the information comes from the Public network, the time to integrate with a new host should be significantly shortened. The IG manufacturer gets a chance to show off its hardware by being in control of the display process. Visualization on a particular IG may be more uniform across applications. Different IGs could be connected to a host without modification to the host for IG replacements or upgrades, etc., allowing IGs to compete on merits not on cost of integration. IGs may differentiate themselves by being able to make use of more network data such as weather information that would be correlated to others using that information from the network.

Another benefit occurs for hosts that do not need to know about targets other than visually, such as the T-38 and C-130. They would not have to implement the portion of the NIU interface that deals with targets, making their integration with the network even easier.

Of course there are some challenges with this architecture. One is that current IGs don't support this. It requires manufacturers to implement network-enabled IGs. Several companies already have "Stealth" products to which a Standard IG interface could be added to get a network-enabled IG. Aechelon, Mak, and Multi-Gen Paradigm are working with AFRL/HEA to implement network-enabled IGs that use the Standard IG interface for communication with the host. The protocol is being refined based on experience with these three integrations, and from input from interested parties.

Network IGs are not "sell-and-forget" devices. If the public network information changes, updates to the IGs NIU may be required. This means that the network IG vendor must be ready to support making these changes for users. This is akin to providing new device drivers for a customer. Once a new NIU driver is created for a new public protocol or Federation Object Model (FOM) then it should be available to all users of that IG. This develop once and distribute procedure is much better than implementing these changes on every host one at a time, and would likely be more cost effective. Perhaps IG manufacturers could work with a third party who specialized in the NIU and allow that third party to provide network protocol updates to users.

One could envision having a network-enabled IG that supports DIS and HLA using the Real-time Platform Reference (RPR) FOM and other FOMs. The more FOMs there are, and the more FOMs are changed, the more difficult it will be for a network IG vendor to keep current. This is also true for all HLA users who want to communicate with each other. Note that an IG usually uses the most basic parts of a FOM that are likely to be quite stable such as things as entities. It will be more likely that things will get added to the FOM rather than changing the basic way an entity is represented by a particular FOM. A single reference FOM for all of the military modeling and simulation community would certainly be helpful for all participants. This is an HLA issue that is just beginning to be addressed.

IG manufacturers could have the opportunity to decide what they would like on the public network to implement visual features, such as weather, dynamic terrain, time of day, etc., and then have direct access to that data rather than relying on integrators to pass that data through their simulations. IGs would be directly accessible to network servers providing information on weather or controllers changing time of day and other environmental conditions. This makes it more likely that all exercise participants would be seeing a more correlated picture. It avoids passing information through the host that it doesn't need, and helps insure that the information an IG needs would be available to it.

Summary and Conclusions

Use of a VIU to translate host information into the format required by an IG provides many benefits including:

- reduced software maintenance
- reduced test time
- reduced debug time
- reduced configuration control problems
- improved upgradability
- improved consistency between hosts
- upgrades that apply to all hosts using the VIU
- fixes that apply to all hosts using the VIU
- automatic integration with any host that uses the NIU
- easier integration with new hosts.

Making the host interface to the VIU the same for all hosts is the key. Using the VIU in conjunction with an NIU as described allows novel capabilities such as the ability to be a "Stealth," which may be used for debrief or demonstration at a later time from logged network data. This significant improvement may be had while still using any existing IG without modifications. As new IGs are supported by the VIU, all hosts get access to those IGs without making any modifications to their host code. By using the Host to NIU interface as the interface for the VIU, a new host only need implement one interface rather than two. This makes the integration quicker and simpler. The VIU concept can be implemented without using the AFRL/HEA NIU. Any place that has all the information generally used by the visual in a common format would be a candidate place to connect a VIU. The backend API of any commercial "NIU" function is a likely place that a VIU may be connected and implemented. The VIU may be used to translate from host formats into the Standard IG interface formats proposed for use with "network-enabled" IGs.

The change from standalone hosts to distributed simulations connected over a public network has changed the way that IGs and hosts relate. With most of the information about the virtual world existing on some type of public DIS/HLA network, it makes sense for IGs to become "network-enabled" for access to that information directly. This gives the IG manufacturers the chance to make use of all network information available. It gives them the opportunity to decide what they would like to be on the public network to implement visual features such as weather. It allows them to directly interact with weather servers and other controllers that may be on the public network. It removes the dependence on a host-to-IG integrator to implement features that an IG manufacturer desires to feature to differentiate it from other IG manufacturers. It simplifies the host IG interface to the most basic of information. That basic information may be standardized easier than a full traditional ICD could be because it need not worry about models and effects that are often tied closely to the particular IG. Such a Standard IG interface is under development at AFRL/HEA in conjunction with Aechelon, M&K Technologies, and Multi-Gen/Paradigm. This standard is open and available to anyone. Comments and suggestions on the standard are welcome. The implementation of network-enabled IGs requires a commitment by the IG vendor to provide upgrades and changes to the NIU portion of the IG to support changes made in the public protocol.

The use of a Standard IG protocol with a networked IG makes the possibility of a "plug-and-play" IG much closer to reality than it is today. This should allow IGs to be changed and upgraded much more easily than is currently possible. By reducing the associated integration costs and risks, it may allow simulators to upgrade their image generators more often as IGs continue to improve rapidly.